S3C44B0X 中文数据手册

6 处理器 Wrapper 和总线优先级

处理器 wrapper 包括一个 cache，写入缓冲器，和 CPU 内核。总线仲裁逻辑决定每个总线占用者的优先权。

处理器 wrapper 有一个 8k 直接的内部存储器。内部存储器可以以 3 种方式采用。第一是 8k 字节的存储空间作为 8k 字节的统一（指令或数据）cache（高速缓冲存储器）。第二，内部存储器可以用作一个 4k 字节的统一 cache 和一个 4k 字节的内部 SRAM。第三，内部存储器可以整个地用作 8k 字节的内部 SRAM。

内部统一（指令或数据）cache 采用 4 种方式与某个 4 个字（16 个字节）的行建立链接结构。它采用一种写穿式（write-through）的原则保持数据的一致性。当在 cache 中找不到相应的内容（称为 cache miss），4 个字的存储内容从外部存储器连续地取得。它采用一种 LRU（最近使用最少）算法来提升命中的比率。统一 cache 通过有区别的方式来处理指令和数据。

内部 SRAM 主要用来减少 ISR 的执行时间。由于内部 SRAM 具有最短的操作时间，因此能够减少 ISR 的执行时间。当然 ISR 在 SRAM 运行也是非常有效率的因为大多数的 ISR 代码都会引起 cache miss。

总线仲裁逻辑可以决定总线占用者的优先级。总线仲裁逻辑支持一种 round-robin 优先级模式和一种固定的优先级模式。同样 LCD_DMA, BDMA, ZDMA, nBREQ（外部总线控制器）之间的优先级可以通过软件来修改。

6．1Cache 的组织

S3C44B0X 的 cache 有一个 8k 字节的 cache 存储器，4 个 RAM 标签和一个 LRU 存储器。

6．2Cache 替换操作

在系统初始化之后，CS 的值被设置为“0000”，这表示 set0, set1, set2 和 set3 cache 存储器是无效的。当某个 cache 填充发生，特定行的 CS 的值改为“0110”，表明只有 set0 是有效的。当并发的 cache 填充发生，……

6．3 cache 禁止操作

S3C44B0X 的 cache 提供整个对 cache 进行使能/禁止的功能。你可以通过设置 SYSCFG 中的 CM 位为 01 或 11 来使能 cache，或者将 SYSCFG[2:1]设置为 00 来禁止 cache。如果 cache 被禁止，指令和数据将一直从外部存储器中获取。S3C44B0X 还可以在 cache 使能模式下提供非 cache 区域，这样使能一些特殊的存储空间操作可以进行，例如 DMA 操作。通过 4 个特殊的寄存器可以指定 2 个非 cache 区域，后面将会介绍到。

当一个存储区域恢复 cache 功能时，保持它与源数据的一致性是非常重要的，因为 cache

存储空间是不会自动刷新的。因此，你必须小心是否 DMA 操作改变了外部存储空间的数据。允许 DMA 操作的存储区域应该设置为非 cache 区域，这样就保证了数据的一致性。S3C44B0X 采用的是写穿式（write-through）的方法来保持 cache 和外部存储器之间数据的一致。

※写穿式(Write Through)

任一从 CPU 发出的写信号送到 Cache 的同时，也写入主存，以保证主存的数据能同步地更新。它的优点是操作简单，但由于主存的慢速，降低了系统的写速度并占用了总线的时间。

6．4 cache 刷新

一个 cache 刷新能够重新使能 cache 操作。当 cache 被禁止，LRU RAM 可以作为一个通常的存储器来操作。Cache 可以通过写入 0 到 LRU RAM 来实现刷新，并使得所有的 cache 数据无效。

注意：cache 刷新必须在 cache 禁止的模式下进行。

6．5 非 cache 区域

S3C44B0X 提供 2 个非 cache 区域。每个要求 2 个 cache 控制区域，控制区域展示了每个非 cache 区域的开始地址和结束地址。在非 cache 区域中，当发生了 cache 读未命中，cache 区域中的数据不会更新。

通常，cache 可以存储整个系统存储器内的任何数据，但是有时它需要非 cache 区域，因为 cache 不能跟踪外部存储设备的数据变化，如果外部设备的数据内容并不总是通过 cpu 对它的读或写操作来更新的，例如允许 DMA 的存储器空间，或者 I/O 口，例如串口等。

非 cache 空间的大小以 4KB 的块为单位增加或减少。结束地址必须指向下一个 4KB 的块。例如，如果非 cache 的区域地址为 0x10000～0x22fff，则 NCACHEBEn 的开始地址值为 0x10，结束地址值为 0x23。

6．6 通过使用 cache 来提高程序的执行速度

1．将经常执行的 ISR 放置到内部 SRAM 执行。
2．让 ISR 不使用 cache，因为大部分的 ISR 代码执行时都会发生 cache 命中失败。而且在 ISR 执行完之后，cache 存储空间不能够再利用，它会立即被主程序代码覆盖。
3．将相关的、互相调用的函数共同执行，这样提高 cache 的命中率。
4．有时，如果将数据区域被设计为非 cache 区域，程序执行速度将会提高，因为大多数变量不会被再利用。对不需要再利用的数据变量，恢复 16 字节的 cache 存储空间的动作将是浪费的。

6．7 内部 SRAM

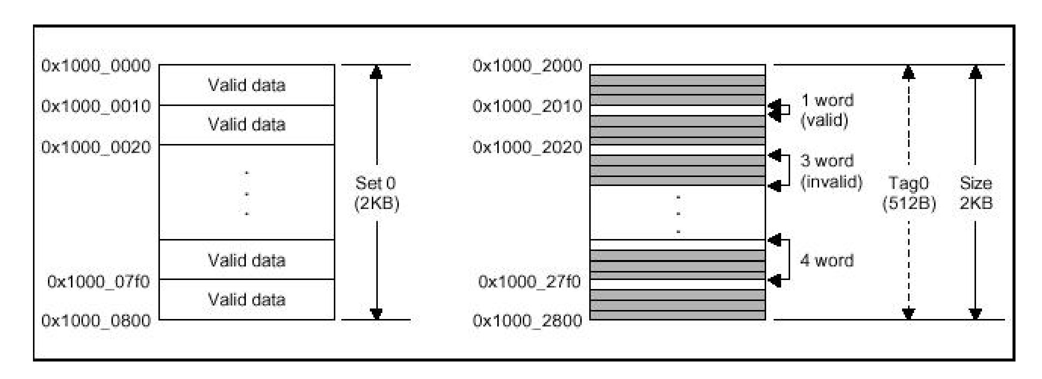
S3C44B0X 具有一个最小 8KB 的 4 组相连 cache 或内部 SRAM。如果内部 SRAM 为 4KB，另外 4KB 内部存储器可以用来作为 2 组相连 cache。内部 SRAM 的存储器操作周期为

1MCLK。

在每一个组的存储空间内，地址是连续增加的，在 TAG/LRU 中的地址按照 16 字节增加。不要对内部地址空间：0x10003004～0x1000300f 进行操作。

区域（组/cache）	存储空间地址	大小
cache set 0	0x10000000 - 0x100007ff	2KB
cache set 1	0x10000800 - 0x10000fff	2KB
cache set 2	0x10001000 - 0x100017ff	2KB
cache set 3	0x10001800 - 0x10001fff	2KB
cache tag 0	0x10002000 - 0x100027f0	512bytes (note)
cache tag 1	0x10002800 - 0x10002ff0	512bytes (note)
cache tag 2	0x10003000 - 0x100037f0	512bytes (note)
cache tag 3	0x10003800 - 0x10003ff0	512bytes (note)
LRU	0x10004000 - 0x100047f0	512bytes (note)

注意：cache 中的 tag3:0 和 LRU 必须以字（32bit）为单位进行读/写。tag3:0 和 LRU 的地址的最低 3 位必须是 0。例如，如果你希望读 tag0 中 128 个项目中的第 2 个，你不应该使用地址 0x10002004，而是 0x10002010。因此，tag0 中的地址为：0x10002000，0x10002010，0x10002020，…，0x100027f0。

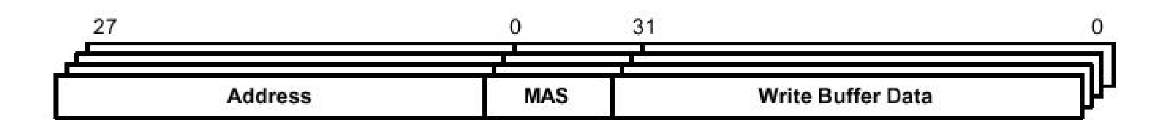


6．8 写缓冲区操作

写缓冲区操作，S3C44B0X 有 4 个写缓冲区寄存器来提高存储器的写性能。当写缓冲器模式使能，CPU 不再将数据直接写入外部存储器而是将数据写入写缓存区。即便是外部总线已经有其它 master 占用，例如 DMA 操作的情况下，也如此。

写缓冲区模块在系统总线没有被别的更高优先级的 master 占用时，将数据写入外部存储器。这样，CPU 的性能就提高了，因为 CPU 不需要一直等到写操作结束。

写缓冲区具有 4 个寄存器。每个寄存器包括一个 32 位数据区域，28 位的地址区域和 2 位的状态区域，如下图所示：



[31:0] Write Buffer Data
将要写到外部存储器中的数据

[1:0] MAS
00 = 8-bit 数据模式
01 = 16-bit 数据模式
10 = 32-bit 数据模式
11 = Not used

[27:0] Address
写入数据的地址

6．9 总线优先级地图

在 S3C44B0X 中，有 7 种总线 master，LCD_DMA, BDMA0, BDMA1, ZDMA0, ZDMA1, Nbreq(外部总线 master)和 CPU wrapper。复位之后，这些总线 master 的优先级排列如下：

1． DRAM 刷新控制器；
2． LCD_DMA；
3． ZDMA0,1
4． BDMA0,1
5． 外部总线 master
6． 写缓冲区
7． Cache 和 cpu

LCD_DMA, ZDMA, BDMA 之间的总线优先级是可编程的，可以通过 SBUSCON 寄存器来设置。如果不考虑 SBUSCON 的设置 CPUwrapper 始终具有最低的优先级。

Round－robin 优先级模式或固定优先级模式是可以选择的。在 round－robin 优先级模式中，已经被服务过依次的总线占有者将具有最低的优先级。通过这种方式，使得所有的 master 都具有相同的优先级。

在固定优先级模式下，每个总线 master 的优先级都写在 SBUSCON 中，SBUSCON 寄存器决定第 1 到第 4 个优先级的总线 master。

6．10 特殊寄存器

对于 CPUwrapper 块（包括 cache，write buffer，和 ARM7TDMI）。SYSCFG 寄存器控制了系统的总体操作。NCACHBE0 和 NCACHBE1 寄存器提供非 cache 区域的设置。

SYSCFG	Bit	Description	Initial State
Reserved	[7]	Reserved to 0	0
Reserved	[6]	Reserved to 0	0
DA(reserved)	[5]	DATA ABORT controls. This bit is recommended to be 0. 0: Enable data abort 1: Disable data abort	0
RSE(reserved)	[4]	Enable read stall option. This bit is recommended to be 0. 0: read stall disable 1: read stall enable (Read stall option: Insert one internal wait cycle when reading data for cache & CPU core.)	0
WE	[3]	This bit determines write buffer enable / disable. Some external devices, which require the minimum writing cycle time, do not operate normally because the period between consecutive writings is shortened the write buffer. 0 = Disable write buffer operation 1 = Enable write buffer operation	0
CM	[2:1]	These two bits determine cache mode 00 = Disable cache (8KB internal SRAM) 01 = Half cache enable (4KB cache, 4KB internal SRAM) 10 = Reserved 11 = Full Cache enable (8KB cache)	00
SE	[0]	Enable stall option. This bit is recommended to be 0. 0:stall disable 1:stall enable (Stall option: Insert one internal wait cycle when a non-sequential address is generated for caching)	1

NCACHBE0：非 cache 区 1 的开始地址和结束地址；

NCACHBE1：非 cache 区 2 的开始地址和结束地址。

SBUSCON	Bit	Description	Initial State
FIX	[31]	0: round-robin priorities 1: fixed priorities	0x1
S_LCD_DMA	[15:14]	Indicates the LCD_DMA bus priority (read only) 00: 1st 01: 2nd 10: 3rd 11: 4th	00
S_ZDMA	[13:12]	Indicates the ZDMA bus priority (read only) 00: 1st 01: 2nd 10: 3rd 11: 4th	01
S_BDMA	[11:10]	Indicates the BDMA bus priority (read only) 00: 1st 01: 2nd 10: 3rd 11: 4th	10
S_nBREQ	[9:8]	Indicates the nBREQ bus priority (read only) 00: 1st 01: 2nd 10: 3rd 11: 4th	11
LCD_DMA	[7:6]	Determines the LCD_DMA bus priority 00: 1st 01: 2nd 10: 3rd 11: 4th	00
ZDMA	[5:4]	Determines the ZDMA bus priority 00: 1st 01: 2nd 10: 3rd 11: 4th	01
BDMA	[3:2]	Determines the BDMA bus priority 00: 1st 01: 2nd 10: 3rd 11: 4th	10
nBREQ	[1:0]	Determines the nBREQ bus priority 00: 1st 01: 2nd 10: 3rd 11: 4th	11